Red Hat Network Satellite 5.5
Guia de [illegible]

Provisionamento e Implantação com o Red Hat Network Satellite
Edição 2

Red Hat Equipe da Documentação

Red Hat Network Satellite 5.5 Guia de [illegible]

Provisionamento e Implantação com o Red Hat Network Satellite
Edição 2

Red Hat Equipe da Documentação

Nota Legal

Resumo

Este documento contém informações e instruções para uso da funcionalidade de provisionamento do kickstart no Red Hat Network Satellite. Para mais sobre o básico do Satellite, veja o Guia do Usuário Satellite.

Índice

Capítulo 1. Introdução

Provisionamento é o processo de configuração de uma máquina virtual ou física para um estado conhecido pré-definido. O Red Hat Network (RHN) Satellite provisiona sistemas que utilizam o processo do *kickstart*. Para usar a função de provisionamento, é necessário obter uma ou mais máquinas *alvo*. As máquinas alvo podem ser físicas ou sistemas bare metal, ou máquinas virtuais. Para usar a função de provisionamento da máquina virtual do RHN Satellite, crie máquinas virtuais utilizando o Xen ou KVM.

Definições

Alguns termos usados nesse livro:

Kickstarting

O processo de instalar um sistema Red Hat de uma maneira automatizada requerendo pouca ou nenhuma intervenção do usuário. Tecnicamente, o *kickstart* se refere ao mecanismo no programa de instalação do Anaconda, o qual permite que você forneça uma descrição concisa do conteúdo e configuração de uma máquina para o instalador, que age em seguida. Tal definição de sistema conciso é referido como um *perfil Kickstart*.

Perfil Kickstart

O arquivo kickstart é um arquivo de texto que especifica todas as opções necessárias para dar um kickstart em uma máquina, incluindo informações de particionamento, configuração de rede, e pacotes a serem instalados. No RHN Satellite, um Perfil Kickstart é um super conjunto das definições kickstart do Anaconda tradicional, pois a implementação do Satellite trabalha baseada nas melhorias do Cobbler para realizar um kickstart. Um Perfil Kickstart presume a existência de uma árvore Kickstart.

Árvore Kickstart

O software e arquivos de suporte necessários para realizar um kickstart em uma máquina. Isto também é chamado de "árvore de instalação". Isto é geralmente uma estrutura de diretório e arquivos vindos da mídia de instalação distribuída em um lançamento específico. Na terminologia do Cobbler, uma Arvore Kickstart, é parte de uma distribuição.

PXE (Preboot eXecution Environment) (Ambiente de execução pré-boot)

Um protocolo de baixo nível que possibilita realizar um kickstart em máquinas bare-metal, (geralmente físicas, ou máquinas *reais*) ligadas sem uma pré-configuração da própria máquina alvo. O PXE conta com um servidor DHCP para informar clientes sobre servidores de bootstrap (para propósito deste documento, instalações do Satellite 5.5). O PXE deve ser suportado no firmware da máquina alvo para ser utilizado. É possível usar a virtualização e reinstalar recursos do Satellite sem o PXE, embora este seja útil para inicializar novas máquinas físicas, ou reinstalar máquinas que não sejam registradas no Satellite.

Cenários de Provisionamento

Os tipos de cenários de provisionamento Suportados pelo RHN Satellite:

Novas instalações

Provisionar um sistema que não teve anteriormente um sistema operacional instalado (também conhecido como instalações *a partir do zero(bare metal)*.

Instalações Virtuais

O Satellite suporta o KVM, hóspedes totalmente virtualizados do Xen, e hóspedes para-virtualizados do Xen.

Re-provisionamento

Ambos sistemas convidados ou físicos, podem ser re-provisionados com o Satellite 5.3.0, se forem registrados na mesma instância do Satellite. Veja o Seção 2.5.2, “Reprovisionar”.

Capítulo 2. Kickstart

2.1. Pacotes requeridos

Se você estiver usando uma distribuição personalizada, você precisará dos seguintes pacotes, que estão disponíveis a partir de qualquer canal Red Hat Network (RHN) **rhn-tools**:

- **koan**
- **spacewalk-koan**

É recomendável clonar um canal **rhn-tools** existente para ter acesso a estes pacotes de seu canal personalizado.

O RHN Satellite Server espera que os arquivos **kernel** e **initrd** estejam em locais específicos dentro da arvore do kickstart. Entretanto, estes locais são diferentes em diferentes arquiteturas. A tabela abaixo explica os diferentes locais:

Tabela 2.1. Arquivos de Distribuição Requeridos por Arquitetura

Arquitetura	kernel	Initial RAM Disk image
IBM System z	***TREE_PATH***/images/kernel.img	***TREE_PATH***/images/initrd.img
PowerPC	***TREE_PATH***/ppc/ppc64/vmlinuz	***TREE_PATH***/images/pxeboot/vmlinux
Todas outras arquiteturas	***TREE_PATH***/images/pxeboot/vmlinuz	***TREE_PATH***/images/pxeboot/initrd.img

2.2. Árvores Kickstart

Você deve ter ao menos uma árvore kickstart instalada no seu Satellite para usar o provisionamento kickstart. Árvores kickstart podem ser instaladas tanto automaticamente ou manualmente.

Procedimento 2.1. Instalando Árvores Kickstart Automaticamente

Para todas as distribuições que possuem um canal base no RHN, as árvores kickstart podem ser instaladas automaticamente. Isto ocorre como parte da sincronização de canal normal via **satellite-sync**.

1. Escolha qual distribuição você gostaria de basear seus kickstarts e localizar este canal de distribuição base, e seus canais de ferramentas RHN.

   Por exemplo, se você quiser usar o Red Hat Enterprise Linux 5 com arquitetura x86, você precisará do canal **rhel-i386-server-5** e seu canal de ferramentas RHN correspondente **rhn-tools-rhel-i386-server-5**.
2. Se você estiver usando um Satellite conectado, sincronize seu Satellite com os servidores Red Hat diretamente utiliando o **satellite-sync**. Se seu servidor Satellite estiver desconectado, você precisará obter descargas de dados desconectado dos servidores Red Hat e sincroniza-los.
3. Sincronizar o canal criará automaticamente uma árvore kickstart correspondente para essa distribuição.

Procedimento 2.2. Instalar Árvores Kickstart Manualmente

Para efetuar o kickstart em uma distribuição personalizada, ou em uma distribuição não suportada pelo

Red Hat ou em uma versão beta do Red Hat Enterprise Linux, você precisará criar uma árvore kickstart manualmente. Você necessitará de uma instalação ISO para a distribuição que você estiver fazendo o kickstart.

1. Copie a instalação ISO para seu servidor Satellite e monte isso ao **/mnt/iso**
2. Copie o conteúdo do ISO para um local personalizado. É recomendado que você crie um diretório dentro do **/var/satellite** para todas suas distribuições personalizadas. Por exemplo, você poderia copiar o conteúdo de uma distribuição beta do RHEL para **/var/satellite/custom-distro/rhel-i386-server-5.3-beta/**
3. Usar a interface web RHN Satellite para criar um canal de software personalizado. Use **Canais** → **Gerenciar Canais de Software** → **Criar Novo Canal** para criar um canal pai com um nome e rótulo apropriado. Para o exemplo usado acima, você poderá usar o rótulo **rhel-5.3-beta**.
4. Mova os pacotes de software do local da árvore para o recém criado canal de software usando o comando **rhnpush**:

```
rhnpush --server=http://localhost/APP -c 'rhel-5.3-beta' \  -d
/var/satellite/custom-distro/rhel-i386-server-5.3-beta/Server/
```

   O sub-diretório dentro da árvore pode ser diferente dependendo de sua distribuição.
5. Uma vez que os pacotes de software tenham sido movidos, eles podem ser deletados do caminho da árvore com o comando **rm**. O pacotes são ainda armazenados no servidor Satellite dentro do canal, e não são mais requeridos pela árvore.

```
rm /var/satellite/custom-distro/rhel-i386-server-5.3-beta/Server/*.rpm
```

Nota

Você pode escolher para deixar os pacotes de software dentro da árvore kickstart. Isto os permitirá serem instalados com o comando **yum** em qualquer momento.

6. Use a interface web RHN Satellite para criar uma distribuição. Use **Sistemas** → **Kickstart** → **Distribuições** → **criar nova distribuição** para criar a distribuição, usando um rótulo apropriado e caminho de árvore inteiro (tal como **/var/satellite/custom-distro/rhel-i386-server-5.3-beta/**). Selecione o canal base que criou anteriormente, e o correto Instalador Generation (tal como **Red Hat Enterprise Linux 5**). Para completar a criação, selecione **Criar Distribuição Kickstart**.
7. Para manter o mesmo software em todos os ambientes e sistemas, o canal filho das Ferramentas RHN de um canal base existente do Red Hat Enterprise Linux pode ser clonado como um filho de um canal base criado recentemente. Clonar um canal filho pode ser feito da seguinte forma:
   a. Na interface da Web do Satellite, clique em **Canais** → **Gerenciar Canais de Software** → **Clonar Canal**
   b. Escolha o Canal que você deseja clonar de uma caixa de menu suspenso **Clonar a partir de:** e escolha o estado do clone.
   c. Clique em **Criar Canal**.
   d. Preencha as informações necessárias e escolha o canal pai sob o qual o canal filho clonado precisa estar.
   e. Clique em **Criar Canal**.

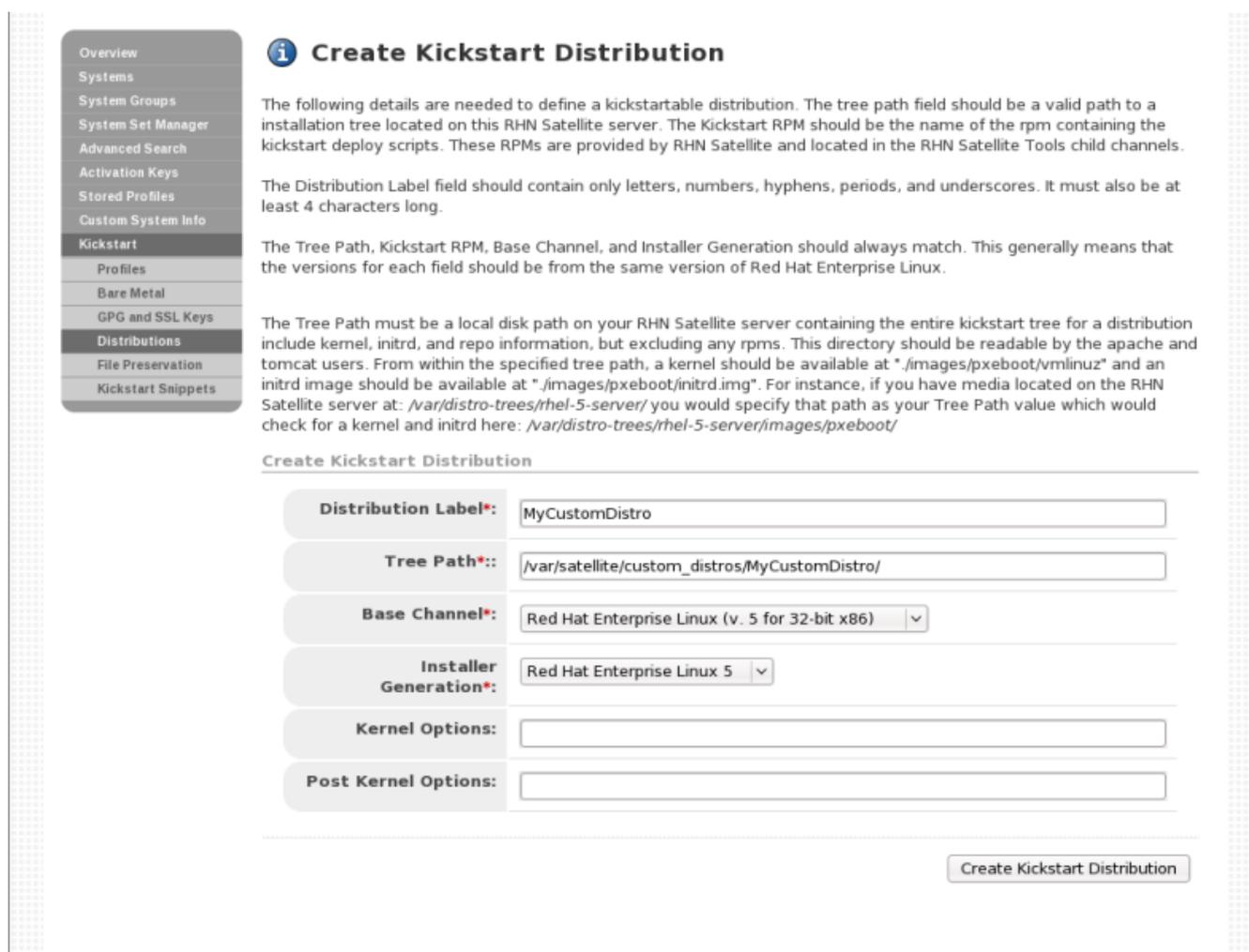



Figura 2.1. Criar uma Distribuição Kickstart

2.3. Perfil de Kickstart

Perfis de Kickstart especificam as opções de configuração a serem usadas na instalação.

Perfis de Kickstart podem ser criados usando uma interface *assistente*, que gera um perfil baseado nas respostas que você dá a uma série de questões. Eles também podem ser criados usando o *método bruto*, que lhe dá controle completo sobre os conteúdos do perfil.

Procedimento 2.3. Criar um Perfil de Kickstart com um Assistente

1. Selecione **Systems** → **Kickstart** → **Criar um Novo Perfil de Kickstart**
2. Forneça um **`Rótulo`** apropriado, e selecione o **`Canal Base`** e **`Árvore Kickstart`** desejados.
3. Selecione o **`Tipo de Virtualização`** desejado. Veja o Tipos de Virtualização para mais informações sobre os tipos de virtualização. Clique **`próximo`** para continuar.
4. Selecione o local para baixar o perfil de Kickstart. Se você estiver usando uma distribuição personalizada, entre o local de sua árvore como uma URI (ambos HTTP e FTP são suportados), caso contrário, use a opção padrão. Clique **`próximo`** para continuar.
5. Entre a senha root e clique **`terminar`** para completar a criação do perfil.
6. Para completar, o perfil de kickstart será criado. Veja o perfil clicando em **`Arquivo Kickstart`**.

Procedimento 2.4. Criar um Perfil Kickstart com o Método Bruto

1. Selecione **Systems** → **Kickstart** → **Upload um Novo Arquivo Kickstart**
2. Forneça um **rótulo** apropriado, e selecione a **distribuição** desejada
3. Selecione o **Tipo de Virtualização** desejada. Veja os Tipos de Virtualização para mais informações sobre tipos de virtualização.
4. Se você tem um perfil kickstart existente, envie o arquivo. Caso contrário, digite o perfil kickstart na caixa de texto **Conteúdo do Arquivo**.

   Aqui segue uma amostra bruta do kickstart que pode ser usada como um ponto de início:

```
install
text
network --bootproto dhcp
url --url http://$http_server/ks/dist/org/1/ks-rhel-i386-server-5
lang en_US
keyboard us
zerombr
clearpart --all
part / --fstype=ext3 --size=200 --grow
part /boot --fstype=ext3 --size=200
part swap --size=1000   --maxsize=2000
bootloader --location mbr
timezone America/New_York
auth --enablemd5 --enableshadow
rootpw --iscrypted $1$X/CrCfCE$x0veQO88TCm2VprcMkH.d0
selinux --permissive
reboot
firewall --disabled
skipx
key --skip

%packages
@ Base

%post
$SNIPPET('redhat_register')
```

5. O RHN Satellite Server não lida com a distribuição especificada como a ***url*** no kickstart, então você precisará incluir a opção ***url --url*** no seu perfil, similar ao seguinte:

```
url --url http://satellite.example.com/ks/dist/org/1/my_distro
```

   Substituir **my_distro** com o rótulo de distribuição e **1** com a id de sua organização.
6. Perfis de Kickstart brutos usam ***$http_server*** ao invés do nome de host do Satellite. Isto será preenchido automaticamente quando o modelo de kickstart é processado.
7. O snippet (trecho) ***redhat_register*** é usado para tratar o registro.

Figura 2.2. Kickstart Bruto

Tipos de Virtualização

Todos perfis de kickstart têm um tipo de virtualização associado a eles. Esta tabela delineia as diferentes opções:

Tabela 2.2. Tipos de Virtualização

Tipo	Descrição	Usos
`nenhum`	Sem virtualização	Use este tipo para provisionamento normal, sistemas com novas instalações, e instalações virtualizadas que não são Xen or KVM (tal como VMware, ou Virtage)
`Hóspede KVM Virtualizado`	Hóspedes KVM	Use este tipo para provisionar visitantes KVM
`Hóspedes Totalmente Virtualizdos Xen`	Hóspedes Xen	Use este tipo para provisionar visitantes Xen **Nota** Esta opção requer suporte a hardware no host, mas não requer um sistema operacional modificado no hóspede.
`Hóspede Para-Virtualizado Xen`	Hóspedes Xen	Use este tipo para provisionar um hóspede virtual com para-virtualização Xen. Para-virtualização é o modo de virtualização mais rápido. Ele requer um sinalizador PAE no sistema CPU, e um sistema operacional modificado. o Red Hat Enterprise Linux 5 suporta visitantes com para-virtualização.
`Virtualização Host Xen`	Xen hosts	Use este tipo para provisionar um host virtual com para-virtualização Xen. Hóspedes e hosts para-virtualizados Xen são suportados, se o hardware for compatível.

Perfis Kickstart criados para serem usados como hosts Xen devem incluir o pacote **`kernel-xen`** na seção ***%packages***.

Perfis Kickstart criados para serem usados como hosts KVM devem incluir o pacote **`qemu`** na seção ***%packages***.

Sistemas totalmente virtualizados podem requerer que o suporte à virtualização seja ativado no menu BIOS do computador.

Nota

Para mais informações sobre o kickstart, veja o capítulo *Instalações Kickstart* no *Guia de Instalação Red Hat Enterprise Linux*.

2.4. Modelação (Templating)

Modelação do kickstart permite que você inclua variáveis, snippets (trechos), e declarações de controle de fluxo tais como declarações **`for`** loops e **`if`** nos seus arquivos kickstart. Isto é feito usando a ferramenta **`cheetah`**.

Existem diversas razões pelas quais a modelação pode ser útil:

- Reusar uma determinada seção de um kickstart, tal como a seção de particionamento de disco, entre múltiplos kickstarts.
- Realizando algumas ações **`%post`** consistentemente através de múltiplos kickstarts.
- Definir um snippet(trecho) através de funções de múltiplos servidores, como um servidor DNS, servidor proxy, e servidor web. Por exemplo, um servidor web poderá ter o seguinte trecho definido:

  ```
  httpd
  mod_ssl
  mod_python
  ```

  Para criar um perfil de servidor web, inclua o snippet de servidor web na seção **`%package`** de seu arquivo de kickstart. Para um perfil ser tanto servidor web e servidor proxy, inclua ambos os trechos no pacote da seção. Para adicionar outro pacote ao trecho do servidor web, por exemplo **`mod_perl`**, atualize o trecho, e todos os perfis que estão usando esse trecho serão atualizados dinamicamente.

Variáveis

A modelação permite que você defina uma variável a ser usada através de todo o arquivo de kickstart. Variáveis estão sujeitas a uma forma de herança que as permitem ser configuradas a um nível e substituir os níveis abaixo. Então, se uma variável é definida ao nível de sistema, ela substituirá as mesmas variáveis definidas nos níveis de perfil ou árvore kickstart. Da mesma maneira, se uma variável é definida a um nível de perfil, ela substituirá as mesmas variáveis definidas ao nível da árvore kickstart.

Nota

Note que as variáveis da árvore kickstart não podem ser definidas para árvores kickstart geradas automaticamente, como aquelas criadas quando você realiza uma sincronização satellite.

Snippets (Trechos)

Snippets reusam pedaços de código entre múltiplos modelos de kickstart. Eles podem abranger muitas linhas, e incluir variáveis. Eles podem ser incluídos em um perfil de kickstart usando o texto **`$SNIPPET('nome_do_snippet')`**. Você pode fazer um snippet para uma lista de pacotes, para um script de **`%post`**, ou para qualquer texto que estaria normalmente incluído num arquivo kickstart.

Para administrar snippets navegue até **Sistemas** → **Kickstart** → **Kickstart Snippets**.

A página **`Kickstart Snippets`** exibe diversos snippets padrões que não podem ser editados, mas estão disponíveis para uso a qualquer organização. Snippets padrões podem ser usados em kickstarts que foram tanto escritos ou enviados ao servidor RHN Satellite. Snippets padrões são armazenados no sistema de arquivos do servidor RHN Satellite, no **`/var/lib/cobbler/snippets/`**. Há um modelo do assitente kickstart localizado no **`/var/lib/rhn/kickstarts/wizard/`**, que explica os diferentes snippets padrões e como eles são usados.

O snippet **`redhat_register`** é um trecho padrão que é usado para registrar máquinas a um servidor RHN Satellite como parte do kickstart. Ele usa uma variável chamada **`redhat_management_key`** para registrar a máquina. Para usar o snippet, configure a variável **`redhat_management_key`** tanto no sistema, perfil ou nível de distribuição e então adicione **`$SNIPPET('redhat_register')`** a uma seção **`%post`** do kickstart. Qualquer assistente de estilo kickstart que são gerados pelo servidor RHN Satellite já incluirão este snippet na seção **`%post`**.

A aba **`Snippets Personalizados`** permite a você ver e editar snippets criados para usar em sua organização. Novos snippets podem ser criados clicando em **`criar novo snippet`**. Snippets personalizados são armazenados no diretório **`/var/lib/rhn/kickstarts/snippets/`**. O RHN Satellite Server armazena snippets para diferentes organizações em diferentes diretórios, então snippets personalizados serão guardados com um nome de arquivo similar ao seguinte, onde ***1*** é o ID da organização:

```
$SNIPPET('spacewalk/1/snippet_name')
```

Para determinar o texto para usar e inserir o snippet no kickstart, veja a coluna **`Snippet Macro`** na lista de snippets, ou na página de detalhes do snippet.

Nota

Snippets existem num nível global e não compartilham as mesmas estrutura de heranças como as variáveis. Entretanto, variáveis podem ser usadas dentro dos snippets para mudar o modo que eles se comportam quando diferentes sistemas solicitarem um kickstart.

Figura 2.3. Kickstart Snippets

Escapar Caracteres Especiais

Os carácteres ***$*** e ***#*** são usados durante modelação para especificar variáveis e controle de fuxo. Se você necessita desses carácteres para qualquer outro propósito num script, eles terão de ser escapados para que eles não sejam reconhecidos como variáveis. Isto pode ser feito de diversas maneiras:

- Colocando uma barra invertida (`\`) antes de qualquer instância de ***$*** ou ***#*** que você quer que seja ignorada durante a modelação.
- Envolva o script inteiro com **`#raw ... #end raw`**

  Todos scripts **`%pre`** e **`%post`** criados usando o assistente de estilos kickstart são envolvidos com ***`#raw...#end raw`*** por padrão. Isto pode ser mudado usando a caixa de seleção **`Modelo`** disponível quando editar um script **`%post`** or **`%pre`**.
- Inclua ***`#errorCatcher Echo`*** na primeira linha do snippet.

Exemplo 2.1. Escapando Carácteres Especiais nos modelos

Este exemplo descreve como escapar carácteres especiais nos modelos de kickstart.

O seguinte script bash necessita ser inserido na seção **%post**:

```
%post
echo $foo > /tmp/foo.txt
```

Sem que o ***$*** seja escapado, o processador de modelação tentará encontrar a variável chamada **$foo** e falharia porque ***foo*** não existe com uma variável.

A maneira mais simples de escapar o ***$*** é usando a barra inversa (****):

```
%post
echo \$foo > /tmp/foo.txt
```

Isto fará que ***\$foo*** seja renderizado como ***$foo***.

Um segundo método é envolver o script bash inteiro em ***#raw ... #end raw***, como a seguir

```
%post
#raw
echo $foo > /tmp/foo.txt
#end raw
```

O método final é incluir ***#errorCatcher Echo*** na primeira linha do modelo de kickstart. Isto instrui o processador de modelação para ignorar qualquer variáveis que não existem e as mostre como texto. Esta opção já está incluída no assistente de estilos do kickstart, e pode ser incluída em qualquer kickstart bruto criado manualmente.

2.5. Fazer um Kickstart em uma Máquina

2.5.1. Fazer Kickstart de uma Nova Instalação

Quando uma máquina não possui um sistema operacional existente ou possui um sistema operacional incorreto instalado, é referida em inglês como uma *bare metal machine* (algo como máquina zerada). Existem três maneiras de provisionar uma máquina a partir do zero:

- Mídia de Instalação de Sistema Operacional Padrão
- PXE boot

Procedimento 2.5. Inicializar a partir de uma Mídia de Instalação

1. Insira a mídia de instalação na máquina. A mídia deve corresponder ao kickstart que você pretende usar. Por exemplo, se o kickstart estiver configurado para usar a árvore kickstart **ks-rhel-i386-server-5-u2**, use a mídia de instalação i386 do RedHat Enterprise Linux 5.2.
2. Ative o kickstart durante a inicialização fornecendo este comando:

```
linux ks=http://satellite.example.com/path/to/kickstart
```

3. Depois que o sistema inicializar, baixe o kickstart e instale-o automaticamente.

Procedimento 2.6. PXE Booting

Para ser capaz de realizar um PXE boot, cada sistema que você tem deve suportar PXE booting a nível de BIOS. Quase todos hardwares recentes são capazes disso. Além disso, você deve ter um servidor DHCP, mesmo se seus sistemas estiverem para serem configurados estaticamente depois da instalação.

1.

   Importante

   Caso o DHCP seja implantado em outro sistema na rede, você precisará acesso administrativo para acessar o servidor DHCP para editar o arquivo de configuração DHCP.

   Se suas máquinas residem em múltiplas redes, você terá de certificar-se que todas suas máquinas podem se conectar ao servidor DHCP. Isto pode ser feito através de multi-homing de seu servidor DHCP (usando tanto uma VLAN real ou trunked) e configurando quaisquer roteadores ou switches para passar o protocolo DHCP através dos limites de rede.

   Configure seu servidor DHCP para que então aponte para o servidor PXE ajustando o endereço ***next-server*** para sistemas que você quer gerenciar pelo RHN Satellite.

   Para utilizar hostnames ao realizar a instalação, configure o servidor DHCP para apontar ao domínio e os endereços IP, incluindo estas linhas:

   ```
   option domain-name DOMAIN_NAME;
   option domain-name-servers IP_ADDRESS1, IP_ADDRESS2;
   ```

2. No servidor DHCP, mude para usuário root e abra o arquivo **/etc/dhcpd.conf**. Anexe uma classe nova com as opções para realizar uma instalação PXE boot.

   ```
   allow booting;
   allow bootp;
   class "PXE" {
     match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";
     next-server 192.168.2.1;
     filename "pxelinux.0";
   }
   ```

   Esta classe realizará as seguintes ações:

   a. Habilita a inicialização da rede com o protocolo **bootp**
   b. Crie uma classe chamada ***PXE***. Se um sistema estiver configurado para ter primeiro o PXE na sua prioridade de inicialização, se identificará como **PXEClient**.
   c. O servidor DHCP direciona o sistema ao servidor Cobbler no endereço de IP ***192.168.2.1***
   d. O servidor DHCP se refere ao arquivo de imagem de boot **/var/lib/tftpboot/pxelinux.0**

3. Configure o Xinetd. O Xinetd é um daemon que gerencia uma suíte de serviços incluindo TFTP, o servidor FTP usado para transferir a imagem de boot a um cliente PXE.

   Habilite o Xinetd usando o comando **chkconfig**:

   ```
   chkconfig xinetd on
   ```

   Alternativamente, mude para usuário root e abra o arquivo **/etc/xinetd.d/tftp**. Encontre a

linha ***disable* = *yes*** e mude para **disable = no**.

4. Inicie o serviço Xinetd para que então o TFTP possa iniciar servindo a imagem de boot **pxelinux.0**:

```
chkconfig --level 345 xinetd on
/sbin/service xinetd start
```

O comando **chkconfig** ativa o serviço **xinetd** para todos os níveis de execução de usuários, enquanto o comando **/sbin/service** ativa o **xinetd** imediatamente.

2.5.2. Reprovisionar

Reinstalar um sistema existente é referido como *reprovisionar*. O reprovisionamento pode ser feito usando a interface web do RedHat Satellite Server, e o sistema usará o mesmo perfil de sistema que tinha antes do reprovisionamento. Isto preservará muitas informações e configurações sobre o sistema.

O reprovisionamento pode ser agendado a partir da aba **Provisionar** enquanto vizualizando um sistema. Para configurar opções adicionais, vá para a página **Configurações Avançadas**, que permite a você configurar detalhes como opções do kernel, informações de rede e sincronização do perfil de pacote. A seção **Opções do Kernel** fornece acesso às opções do kernel que serão usadas durante e as opções do kernel **Após Opções do Kernel** serão utilizadas depois que o kickstart estiver completo e o sistema estiver inicializando pela primeira vez.

Exemplo 2.2. Configurar as Opções do Kernel e Opções Post Kernel

Este exemplo descreve a diferença entre opções do kernel e opções do post kernel no processo de configuração do reprovisionamento.

Para estabelecer uma conexão VNC para monitorar o kickstart remotamente, inclua **vnc vncpassword=PASSWORD** na linha **Kernel Options**.

Se você quer que o kernel do sistema resultante inicialize com a opção kernel **noapic** adicione **noapic** à linha **Post Kernel Options**.

Procedimento 2.7. Preservação de Arquivo

O recurso *Preservação de Arquivo* pode ser usado para evitar que arquivos sejam perdidos durante um reprovisionamento. Este recurso armazena arquivos temporiaramente durante o kickstart e os restaura depois que o reprovisionamento estiver completo.

Nota

Listas de preservação de arquivo estão somente disponíveis nos assistentes de estilo do kickstart, e podem somente ser usadas durante reprovisionamentos.

1. Vá até **Sistemas** → **Kickstart** → **Preservação de Arquivo** → **criar nova lista de preservação de arquivos** e crie uma lista de arquivos a serem preservadas.
2. Vá até **Sistemas** → **Kickstart** → **Perfis** e associe a lista de preservação de arquivo com um kickstart selecionando o perfil desejado.
3. Vá até **Detalhes do Sistema** → **Preservação de Arquivos** e selecione a lista de preservação de arquivos.

2.5.3. Provisionamento de Hóspedes Virtualizados

O provisionamento de Hóspedes Virtuais é suportado no RHN Satellite 5.5 , usando as seguintes tecnologias de virtualização:

- Hóspedes Virtualizados KVM
- Hóspedes Totalmente-Virtualizados Xen
- Hóspedes Para-Virtualizados Xen

Procedimento 2.8. Provisionar um Hóspede Virtualizado

1. Cheque que o sistema de host tenha um sistema de direitos de **`Virtualização`** ou **`Plataforma de Virtualização`**
2. Na página **`Sistemas`**, selecione um host virtual apropriado, então selecione **`Virtualização`** → **Provisionar**
3. Para configurar parâmetros adicionais tais como memória de hóspede e uso de CPU, clique no botão **`Configuração Avançada`**. Permitindo que você configure:
   - Rede: estática ou DHCP
   - Opções do kernel
   - Pacote de sincronização de perfil: quando o kickstart termina, o sistema sincronizará seu pacote de perfil àquele outro sistema ou perfil armazenado.
   - Alocação de memória: RAM (O padrão é de 512MB)
   - Tamanho de disco virtual
   - CPUs Virtuais (O padrão é 1)
   - Virtual bridge: A networking bridge usada para a instalação. O padrão é **`xenbr0`** para provisionar Xen, e **`virbr0`** para KVM.

   Nota

   A networking bridge **`virbr0`** não permitirá contato de rede externo. Se você precisar de contato externo, configure o host para criar uma bridge real. Entretanto, o **`xenbr0`** é uma bridge real, e é recomendado para uso se possível.

   - Caminho de armazenamento virtual: O caminho tanto para um arquivo, Volume Lógico LVM, diretório ou dispositivo em bloco que armazena as informações de disco do hóspede, tais como **`/dev/sdb`**, **`/dev/LogVol00/mydisk`**, **`VolGroup00`**, ou **`/var/lib/xen/images/myDisk`**.
4. Clique **`Agendar Kickstart e Terminar`**

2.5.4. Provisionar através de uma RHN Proxy

O provisionamento pode também ser alcançado usando uma RHN Proxy que foi instalada e registrada para o RHN Satellite.

1. Quando provisionar um hóspede virtual ou fazer um reprovisionamento de um sistema, selecione a proxy desejada da caixa de seleção suspensa **`Select Satellite Proxy`**.
2. Para instalações a partir do zero, substitua o nome de domínio totalmente qualificado (FQDN) RHN Satellite com o da proxy FQDN. Por exemplo se a URL para o arquivo de kickstart é:

```
http://satellite.example.com/ks/cfg/org/1/label/myprofile
```

Então para fazer um kickstart por uma proxy, use:

```
http://proxy.example.com/ks/cfg/org/1/label/myprofile
```

Capítulo 3. Satellites múltiplos

Sincronização Inter-satellite (ISS - Inter-satellite synchronization) permite a você coordenar conteúdo entre Satellites. Este recurso pode ser usado de diversas maneiras, dependendo das necessidades de sua organização. Este capítulo contém uma seção de casos de uso e como melhor configurar o ISS para atender sua organização.

Requerimentos do ISS

Estas são as condições necessárias para poder utilizar o ISS:

- Dois ou mais servidores do RHN Satellite
- Pelo menos um RHN Satellite populado com pelo menos um canal
- Para conexões seguras, cada RHN Satellite slave também precisará de um certificado SSL master RHN Satellite

3.1. Sincronização do Inter-Satellite

Procedimento 3.1. Configurando o Servidor Master

O servidor master é usado para determinar quais arquivos serão sincronizados aos outros satellites.

1. Habilite o recurso de sincronização inter-satellite (ISS). Abra o arquivo **/etc/rhn/rhn.conf**, e adicione ou altere a seguinte linha:

```
disable_iss=0
```

2. No arquivo **/etc/rhn/rhn.conf**, localize a linha **allowed_iss_slaves=**. Por padrão, nenhum satellite slaves são especificados para sincronização. Entre o nome de host de cada servidor satellite slave, separados por vírgulas:

```
allowed_iss_slaves=slave1.satellite.example.org,slave2.satellite.example.org
```

3. Salve o arquivo de configuração, e reinicie o serviço **httpd**:

```
service httpd restart
```

Procedimento 3.2. Configurar Servidores Slaves

Servidores Satellites Slaves são as máquinas que terão seu conteúdo sincronizado ao servidor master.

1. Para tranferir seguramente o conteúdo aos servidores slave, você precisará do certificado **ORG-SSL** do servidor master. Você pode baixar o certificado por HTTP do diretório **/pub/** de qualquer satellite. O arquivo é chamado **RHN-ORG-TRUSTED-SSL-CERT**, mas pode ser renomeado e colocado em qualquer lugar no sistema de arquivos local do slave, tal como o diretório **/usr/share/rhn/**.

2. Vizualize a lista de canais disponíveis para sincronização do servidor master com o seguinte comando. Isto mostrará os canais oficiais Red Hat tão como quaisquer canais personalizados.

```
satellite-sync --iss-parent=master.satellite.example.com --ca-
cert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --list-channels
```

   Substitua ***master.satellite.example.com*** com o nome de host do servidor master.

Procedimento 3.3. Realizando uma Sincronização Inter-Satellite

Uma vez que os servidores master e slave estiverem configurados, você pode realizar a sincronização entre eles.

1. Nos servidores slave, abra o arquivo **`/etc/rhn/rhn.conf`** em seu editor de texto preferido, e adicione o nome de host do servidor master e os detalhes do caminho do certificado SSL:

   ```
   iss_parent      = master.satellite.example.com
   iss_ca_chain    = /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
   ```

2. Inicie a sincronização rodando o comando **`satellite-sync`**:

   ```
   satellite-sync -c your-channel
   ```

Nota

Opções de linha de comando que são fornecidas manualmente com o comando **`satellite-sync`** irão sobrepor qualquer configuração personalizada no arquivo **`/etc/rhn/rhn.conf`**.

3.2. Sincronização Organizacional

O ISS pode também ser usado para importar conteúdo para qualquer organização específica. Isto pode ser feito localmente ou por qualquer sincronização remota. Esta função é útil para um satellite desconectado com múltiplas organizações, onde o conteúdo é recebido através de dump dos canais ou exportação de satellites conectados e então importando a um satellite desconectado. A sincronização organizacional pode ser usada para exportar canais personalizados de satellites conectados. Isso também pode ser usado efetivamente para mover conteúdo entre múltiplas organizações.

A sincronização organizacional segue um conjunto de regras claras para manter a integridade da organização fonte:

- Se o conteúdo fonte pertence a uma organização ***`NULL`*** (que é conteúdo Red Hat), isto fará padrão à organização ***`NULL`***, mesmo se uma organização destino é especificada. Isto certifica que o conteúdo especificado está sempre na organização ***`NULL`*** privilegiada.
- Se uma organização estiver especificada na linha de comando, o conteúdo será importado dessa organização.
- Se nenhuma organização é especificada, então será padrão a organização 1.

A seguir estão três exemplos de cenários onde IDs organizacionais (***`orgid`***) são usados para sincronizar satellites:

Exemplo 3.1. Importar Conteúdo do Master para o Satellite Slave

Este exemplo importa conteúdo do master para o satellite slave:

```
satellite-sync --parent-sat=master.satellite.example.com -c channel-name --
orgid=2
```

Exemplo 3.2. Importar Conteúdo de um Dump Exportado de uma Organização

Este exemplo importa conteúdo de um dump exportado de uma organização específica:

```
$ satellite-sync -m /dump -c channel-name --orgid=2
```

Exemplo 3.3. Importar conteúdo de um Hosted RedHat Network

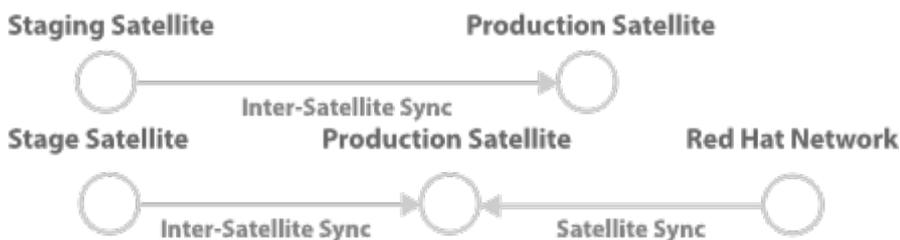
Este exemplo importa conteúdo de um Hosted Red Hat Network (considerando que o sistema está registrado e ativado):

```
$ satellite-sync -c channel-name
```

3.3. Casos de Uso do ISS

O ISS pode ser usado de diversas maneiras, dependendo das necessidades de sua organização. Esta seção fornece exemplos de como você pode escolher o uso do ISS, e os métodos para configurar e operar nesses casos.

Exemplo 3.4. Staging Satellite (Satellite em teste)

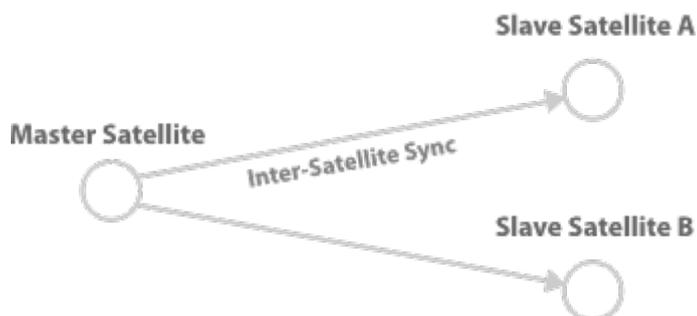
Este exemplo usa um satellite como um *staging satellite (teste)* para preparar o conteúdo e realizar controle de qualidade dos pacotes para ter certeza que eles estão corretos para uso em produção. Quando o conteúdo é aprovado para produção, o satellite em produção pode sincronizar o conteúdo do satellite em teste.

1. Rode o comando **`satellite-sync`** para sincronizar os dados com **`rhn_parent`** (normalmente RedHat Network Hosted):

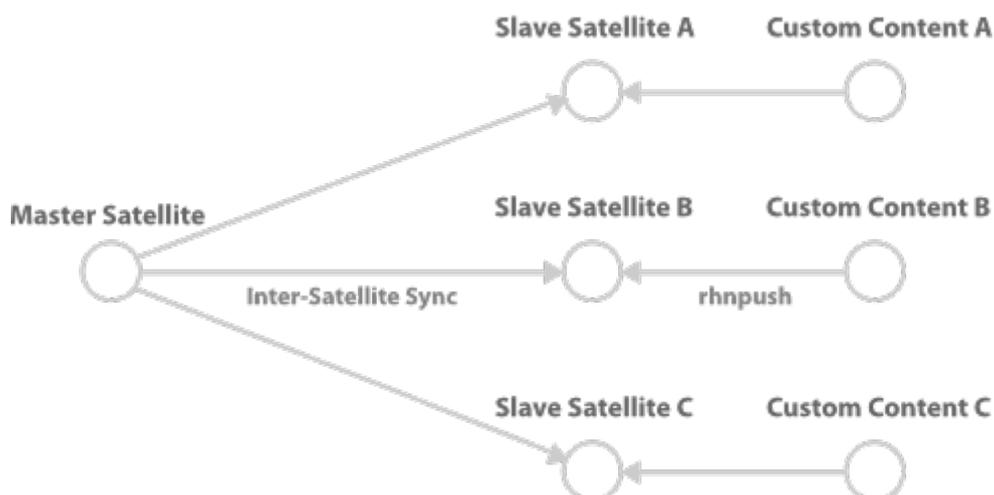
   ```
   satellite-sync -c your-channel
   ```

2. Rode o seguinte comando para sincronizar os dados de um staging server (servidor em teste):

   ```
   satellite-sync --iss-parent=staging-satellite.example.com -c custom-channel
   ```

Exemplo 3.5. Slaves sincronizados

Neste exemplo, o satellite master fornece dados diretamente aos slaves e mudanças são sincronizadas regularmente.

Exemplo 3.6. Conteúdo Personalizado do Slave

Este exemplo usa o satellite master como um canal de desenvolvimento, do qual o conteúdo é distribuído para todos os satellites slaves. Alguns dos satellites slave possuem conteúdos extra que não estão presentes nos canais do satellite master. Estes pacotes são preservados, mas todas alterações do satellite master são sincronizadas aos slaves.

Exemplo 3.7. Sincronia bi-direcional

Neste ambiente, dois servidores RHN Satellite agem como master um do outro e podem sincronizar conteúdo entre eles.

1. Tenha certeza que ambos Satellites possam compartilhar certificados SSL.
2. No primeiro satellite, abra o arquivo **`/etc/rhn/rhn.conf`** e configure a opção **`iss_parent`** para apontar o nome de host ao segundo Satellite.
3. No segundo Satellite, abra o arquivo **`/etc/rhn/rhn.conf`** e configure a opção **`iss_parent`** para apontar ao nome de host do primeiro Satellite.

Capítulo 4. Comando e Métodos API Avançados

4.1. O API XML-RPC

RHN Satellite 5.5 suporta o provisionamento utilizando o XML-RPC API.

Os seguintes métodos API são usados para perfis de kickstart e manutenção de árvore:

Tabela 4.1. Métodos XML-RPC

XML-RPC Namespace	Usagem
`kickstart`	Cria, importa e deleta perfis de kickstart. Também usado para listar árvores de kickstart e perfis disponíveis.
`kickstart.tree`	Cria, renomeia, atualiza e deleta árvores de kickstart.
`kickstart.filepreservation`	Lista, cria e deleta listas de preservação de arquivos que podem ser associadas ao perfil de kickstart. Uma vez que a lista de preservação de arquivo foi criada, ela pode ser associada ao perfil de kickstart chamando o método API **kickstart.profile.system.add_file_preservations**
`kickstart.keys`	Liste, crie e delete chaves criptográficas (GPG/SSL) que podem ser associadas a diferentes perfis de kickstart. **Nota** Uma vez que a chave criptográfica foi criada, ela pode ser associada ao perfil de kickstart chamando o método API **kickstart.profile.system.add_keys**
`kickstart.profile`	Manipula faixas de IP, altera a árvore de kickstart e os canais filhos, baixa os arquivos de kickstart associados com o perfil, manipula opções avançadas, manipula opções personalizadas, e adiciona pre- e post- scripts ao um perfil de kickstart.
`kickstart.profile.keys`	Lista, adiciona (associa), e remove (disassocia) chaves de ativação associadas a um perfil kickstart.
`kickstart.profile.software`	Manipula a lista de pacotes associados a um perfil kickstart.
`kickstart.profile.system`	Gerencia preservações de arquivos, gerencia chaves criptográficas, habilita/desabilita gerenciamento de configurações e comandos remotos, configura esquemas de partição, e configura informações locais associadas a um perfil kickstart dado.

Os seguintes métodos API são usados para re-provisionar um host e agendar instalações nos hóspedes:

- **system.provision_system**
- **system.provision_virtual_guest**

Para mais informações nas chamadas de API veja a documentação de API disponível em **`https://satellite.example.com/rpc/api`**

4.2. Cobbler

O RHN Satellite usa o Cobbler para provisionamento. Quando os perfis de kickstart, árvores(distribuições), e sistemas de provisionamento são atualizados no RHN Satellite, eles são sincronizados à instância do Cobbler no RHN Satellite. Isto significa que o Cobbler pode ser usado diretamente para gerenciar provisionamento.

A tabela seguinte descreve os comandos do Cobbler:

Tabela 4.2. Comandos do Cobbler

Comando	Usagem
`cobbler profile list`	Execute este comando no host do RHN Satellite para exibir uma lista de perfis
`cobbler distro list`	Exibe uma lista de árvores de kickstart, kernels, discos RAM, e outras opções
`cobbler system list`	Exibe uma lista de registros do sistema, criados quando um kickstart é agendado
`cobbler profile report --name=profile-name` or **`cobbler system report --name=system-name`**	Exibe um resultado mais detalhado sobre um objeto específico
`cobbler profile edit --name=profile-name --virt-ram=1024`	Edita vários parâmetros. Este exemplo alocará a cada instalação virtualizada de um perfil dado, 1GB de RAM
`cobbler system edit --name=system-name --netboot-enabled=1`	Força um sistema a ser reinstalado na próxima inicialização
`cobbler system edit --name=system-name --profile=new-profile-name --netboot-enabled=1`	Atribui um sistema a um novo perfil para reinstalação
`cobbler system find --profile=profile-name`	Lista todos os sistemas atribuídos a um perfil
`cobbler system find --profile="abc" | xargs -n1 --replace cobbler system edit \ --name={} --profile="def" --netboot-enabled=1`	Atribui todos sistemas atualmente configurados ao perfil ***`abc`*** ao perfil ***`def`*** e os reinstala na próxima vez que eles reinicializam
`cobbler profile edit --name=profilename --kopts="variablename=3" --in-place`	Configura uma variável de modelação adicional em um perfil sem modificar qualquer das outras variáveis
`cobbler system edit --name=systemname --kopts="selinux=disabled asdf=jkl"`	Atribui diversas variáveis a um registro de sistema, e desconsidera quaisquer variáveis antigas que possam ser ajustadas
`cobbler profile find --name="*webserver*" | xargs -n1 --replace cobbler profile edit --name={} --profile="RHEL5-i386"`	Configura todas as novas instalações de qualquer perfil contendo ***`webserver`*** como uma série para usar um perfil chamado ***`RHEL5-i386`***

Outras configurações do Cobbler

Existem somente poucas configurações do Cobbler que deveriam ser alteradas no **/etc/cobbler/settings** diretamente. A opção **pxe_just_once** é uma delas (descritas no Procedimento 4.3, "Configurando o Cobbler para usar o PXE"). A opção **server** pode também ser alterada para refletir o endereço ou nome de host do RHN Satellite Server.

Depois de alterar o **/etc/cobbler/settings**, execute o seguinte comando para captar as mudanças:

```
/sbin/service cobblerd restart
cobbler sync
```

Importante

Não ajuste qualquer outra configuração no **/etc/cobbler/settings**. O RHN Satellite requer que este arquivo permaneça numa certa configuração, determinada pelo instalador do RHN Satellite. Da mesma forma, o arquivo **/etc/cobbler/modules.conf**, que controla fontes de autenticação, deveriam permanecer como criados pelo instalador do RHN Satellite. Particularmente, o módulo de autenticação deve permanecer como **authn_spacewalk** e não é alterável.

Procedimento 4.1. Configurando o SELinux para uso com o Cobbler

O suporte a SELinux e um firewall seguro são instalados por padrão com o Red Hat Enterprise Linux. Para propriamente configurar um servidor Red Hat Enterprise Linux para usar o Cobbler, o SELinux deve estar configurado para permitir conexões para e a partir do servidor Cobbler.

1. Para habilitar o SELinux para suporte ao Cobbler, configure o SELinux Boolean para permitir os componentes de serviço web HTTPD, usando o seguinte comando:

   ```
   setsebool -P httpd_can_network_connect true
   ```

   O comutador **-P** é essencial, já que habilita a conexão HTTPD persistentemente por todas reinicializações de sistemas.
2. Configure as regras de contexto do arquivo SELinux para o TFTP servir o arquivo de imagem de inicialização, usando os seguintes comandos no servidor Cobbler:

   ```
   semanage fcontext -a -t public_content_t "var/lib/tftpboot/.*"
   ```

3. O IPTables devem ser configurados para permitir entradas e saídas de tráfico de rede no servidor Cobbler.

   Se você tem um conjunto de regras de firewall usando iptables, adicione as seguintes regras para abrir as portas relacionadas do Cobbler, como a seguir:

   Para TFTP:

   ```
   /sbin/iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 69 -j
   ACCEPT
   /sbin/iptables -A INPUT -m state --state NEW -m udp -p udp --dport 69 -j
   ACCEPT
   ```

Para HTTPD:

```
/sbin/iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j
ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j
ACCEPT
```

Para o Cobbler:

```
/sbin/iptables -A INPUT -m state --state NEW -m udp -p udp --dport 25150 -j
ACCEPT
```

Para o Koan

```
/sbin/iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 25151 -j
ACCEPT
```

4. Salve as configurações de firewall:

```
/sbin/iptables-save
```

5. Assegure-se que os arquivos de configuração estão todos sincronizados executando o seguinte comando:

```
cobbler sync
```

6. Inicie o servidor Satellite:

```
/usr/sbin/rhn-satellite start
```

Aviso

Não inicie ou pare o serviço **`cobblerd`** independente do serviço do Satellite, pois poderá causar erros e outros problemas.
Sempre use **`/usr/sbin/rhn-satellite`** para iniciar ou parar o RHN Sattelite.

Procedimento 4.2. Configurando Registros de Sistema do Cobbler

Registros de sistema do Cobbler são objetos dentro do Cobbler que acompanham um sistema e são associados ao perfil de kickstart. Para realizar um kickstart PXE, um perfil Kickstart Satellite deve estar atado aos registros do sistema do Cobbler para as máquinas que você pretende fazer o kickstart.

1. Vá até **Detalhes do Sistema** → **Provisionamento** para cada sistema e selecione o perfil de kickstart a ser associado.
2. Clique **`Criar Registro de Sistema do Cobbler`** para fazer a associação.
3. A associação permanecerá no lugar indefinitivamente a menos que você configure a opção **`pxe_just_once`** para verdadeira para qualquer máquina dada. Neste caso a associação será rompida após um kickstart bem sucedido. Veja o Procedimento 4.3, “Configurando o Cobbler para usar o PXE” para mais informações sobre esta configuração.

Procedimento 4.3. Configurando o Cobbler para usar o PXE

Cobbler está definido para gerar configurações do PXE por padrão, mas para obter o melhor fluxo de

trabalho do PXE possível no BIOS, a opção de configuração do **pxe_just_once** pode ser ajustada.

1. Muitas vezes, a ordem da BIOS será configurada para fazer o boot do PXE ocorrer primeiro. Isto significa que o sistema não irá inicializar do disco local ao menos que o servidor PXE o instrua a fazer remotamente. Esta configuração pode criar um *boot loop*, onde o sistema reinstala continuamente.

   Para prevenir boot loops, abra o arquivo **/etc/cobbler/settings** e adicione a seguinte linha:

   ```
   pxe_just_once: 1
   ```

   Esta configuração adiciona uma macro ***$kickstart_done*** no modelo de kickstart, que diz ao sistema para inicializar localmente depois de ter completado a instalação, ao invés de inicializar a partir da rede.

2. Se você incluir a configuração ***pxe_just_once: 1***, e quiser reinstalar o sistema mais tarde, você precisará alternar a marca ***netboot-enabled*** no sistema. Isto pode ser feito usando tanto a interface web RHN Satellite, ou no Cobbler diretamente. Quando o sistema reinicializar na próxima vez, realizará uma instalação PXE, e então retornará para a inicialização a partir do disco local até que a marca seja restabelecida.

   Se a BIOS estiver configurada para inicializar a partir de discos rígidos locais primeiro, não há necessidade de ter o ***pxe_just_once*** habilitado. Entretanto, para reprovisionar o sistema usando PXE, será necessário zerar o MBR (master boot record).

Nomeando Convenções

Para ajudar a manter a data sincronizada entre o RHN Satellite e o Cobbler, o RHN Satellite usa convenção de nomes para distribuição e perfis. Esta convenção de nomes são importantes se você interagir com o Cobbler usando a interface de linha de comando.

Distribuições

$tree_name:$org_id:$org_name (if manually created)

$tree_name (if synchronized by RHN Satellite)

Perfis

$profile_name:$org_id:$org_name

Importante

Não altere nomes que foram automaticamente gerados pelo RHN Satellite. Se um nome é mudado, o RHN Satellite não poderá manter estes ítems.

Nota

Por razões de troubleshooting o Cobbler autentica dados nos logs do RHN Satellite e no arquivo **/var/log/cobbler/**.

4.3. Koan

O utilitário **koan** (kickstart over a network) permite ao Satellite ser acessado remotamente de hosts que já foram provisionados. O Koan permite que você realize provisionamento de kickstart, criar hóspedes virtuais (em hosts virtuais), e possa listar os kickstarts disponíveis do host Satellite. Ele está disponível no pacote **koan**.

Tabela 4.3. Comandos do Koan

Comando	Usagem
man koan	Leia a página man do **koan**
koan --replace-self --server=satellite.example.org --profile=profile-name or **koan --replace-self --server=satellite.example.org --system=system-name**	Reprovisionar um sistema existente. Reinicie depois de executar este comando para instalar o novo sistema operacional. Isto também pode ser usado com kickstarts de atualização (por exemplo, para atualizar um grande número de máquinas de uma versão do Red Hat Enterprise Linux para a próxima).
koan --virt --server=satellite.example.org --profile=profile-name or **koan --virt --server=satellite.example.org --system=system-name**	Provisionar um hóspede virtual
koan --list=profiles --server=satellite.example.org or **koan --list=systems --server=satellite.example.org**	Comanda o Cobbler para exibir uma lista de perfis ou sistemas disponíveis para instalação remota

Nota

Por razões de troubleshooting, observe que o Koan autentica dados em **/var/log/koan/**.

Capítulo 5. Solução de problemas

5.1. Interface web

P: **Estou tendo problemas com a interface de usuário do RHN Satellite. Quais arquivos de log eu devo checar?**

R: Se você tem erros vizualizando, agendando, ou trabalhando com kickstarts na interface de usuário do RHN Satellite Server, cheque o arquivo de log **`/var/log/tomcat5/catalina.out`**.

Para todos os outros erros da interface de usuário, cheque o arquivo de log **`/var/log/httpd/error_log`**.

5.2. Anaconda

P: **Estou tendo um erro que diz `Error downloading kickstart file (Erro ao baixar o arquivo de kickstart)`. Qual é o problema e como conserto isso?**

R: Este erro é geralmente o resultado de um problema de rede. Para localizar o problema, rode o comando **`cobbler check`**, e leia o resultado, que deve mostrar algo como:

```
# cobbler check
The following potential problems were detected:
#0: reposync is not installed, need for cobbler reposync, install/upgrade
yum-utils?
#1: yumdownloader is not installed, needed for cobbler repo add with --rpm-
list parameter, install/upgrade yum-utils?
#2: The default password used by the sample templates for newly installed
machines (default_password_crypted in /etc/cobbler/settings) is still set to
'cobbler' and should be changed
#3: fencing tools were not found, and are required to use the (optional) power
management features. install cman to use them
```

Se o **`cobbler check`** não fornecer uma resposta, cheque o seguinte:

- Verifique se **`httpd`**está sendo executado: **`service httpd status`**
- Verifique se o **`cobblerd`** está sendo executado: **`service cobblerd status`**
- Verifique se você pode pegar o arquivo de kickstart usando **`wget`** de um host diferente:

```
wget http://satellite.example.com/cblr/svc/op/ks/profile/rhel5-i386-
u3:1:Example-Org
```

P: **Estou tendo um erro num pacote de instalação que diz `The file` *`chkconfig-1.3.30.1-2.i386.rpm`* `cannot be opened.`. Qual é o problema e como eu conserto isso?**

R: Máquinas clientes pegarão conteúdo do RHN Satellite baseados no parâmetro ***`--url`*** no kickstart. Por exemplo:

```
url --url http://satellite.example.com/ks/dist/ks-rhel-i386-server-5-u3
```

Se você receber erros do Anaconda dizendo que não pode encontrar imagens ou pacotes, cheque de que a URL no kickstart gere uma reposta ***`200 OK`***. Você pode fazer isso tentando wget no arquivo localizado na URL:

```
wget http://satellite.example.com/ks/dist/ks-rhel-i386-server-5-u3
--2011-08-19 15:06:55--  http://satellite.example.com/ks/dist/ks-rhel-i386-
server-5-u3
Resolving satellite.example.com... 10.10.77.131
Connecting to satellite.example.com|10.10.77.131|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: `ks-rhel-i386-server-5-u3.1'
2011-08-19 15:06:55 (0.00 B/s) - `ks-rhel-i386-server-5-u3.1' saved [0/0]
```

Se você obter uma outra resposta além de ***200 OK***, cheque os logs de erro para encontrar qual é o problema. Você pode também checar o arquivo Anaconda real que se foi tentado baixar procurando o arquivo **access_log**:

```
# grep chkconfig /var/log/httpd/access_log
10.10.77.131 - - [19/Aug/2011:15:12:36 -0400] "GET
/rhn/common/DownloadFile.do?url=/ks/dist/ks-rhel-i386-server-
5-u3/Server  /chkconfig-1.3.30.1-2.i386.rpm HTTP/1.1" 206 24744 "-"
"urlgrabber/3.1.0 yum/3.2.19"
10.10.76.143 - - [19/Aug/2011:15:12:36 -0400] "GET /ks/dist/ks-rhel-i386-
server-5-u3/Server/chkconfig-
1.3.30.1-2.i386.rpm HTTP/1.1" 206 24744 "-" "urlgrabber/3.1.0 yum/3.2.19"
10.10.76.143 - - [19/Aug/2011:15:14:20 -0400] "GET /ks/dist/ks-rhel-i386-
server-5-u3/Server/chkconfig-
1.3.30.1-2.i386.rpm HTTP/1.1" 200 162580 "-" "urlgrabber/3.1.0 yum/3.2.19"
10.10.77.131 - - [19/Aug/2011:15:14:20 -0400] "GET
/rhn/common/DownloadFile.do?url=/ks/dist/ks-rhel-i386-server-
5-u3/Server/chkconfig-1.3.30.1-2.i386.rpm HTTP/1.1" 200 162580 "-"
"urlgrabber/3.1.0 yum/3.2.19"
```

Se as requisições não estão aparecendo no arquivo **access_log**, o sistema pode estar tendo problemas com a configuração de rede. Se as requisições estão aparecendo mas estão gerando erros, cheque os logs de erro.

Você pode também tentar manualmente baixar os arquivos para ver se o pacote está disponível:

```
wget http://satellite.example.com/ks/dist/ks-rhel-i386-server-5-
u3/Server/chkconfig-1.3.30.1-2.i386.rpm
```

5.3. Tracebacks

P: Estou recebendo emails com "WEB TRACEBACK" no campo assunto. O que eu devo fazer sobre isso?

R: Um email de traceback típico é parecido com o seguinte:

```
Subject: WEB TRACEBACK from satellite.example.com
Date: Wed, 19 Aug 2011 20:28:01 -0400
From: RHN Satellite <dev-null@redhat.com>
To: admin@example.com

java.lang.RuntimeException: XmlRpcException calling cobbler.
 at
com.redhat.rhn.manager.kickstart.cobbler.CobblerXMLRPCHelper.invokeMethod(Cobb
lerXMLRPCHelper.java:72)
 at
com.redhat.rhn.taskomatic.task.CobblerSyncTask.execute(CobblerSyncTask.java:76)
 at
com.redhat.rhn.taskomatic.task.SingleThreadedTestableTask.execute(SingleThreaded
TestableTask.java:54)
 at org.quartz.core.JobRunShell.run(JobRunShell.java:203)
 at
org.quartz.simpl.SimpleThreadPool$WorkerThread.run(SimpleThreadPool.java:520)
Caused by: redstone.xmlrpc.XmlRpcException: The response could not be parsed.
 at redstone.xmlrpc.XmlRpcClient.handleResponse(XmlRpcClient.java:434)
 at redstone.xmlrpc.XmlRpcClient.endCall(XmlRpcClient.java:376)
 at redstone.xmlrpc.XmlRpcClient.invoke(XmlRpcClient.java:165)
 at
com.redhat.rhn.manager.kickstart.cobbler.CobblerXMLRPCHelper.invokeMethod(Cobb
lerXMLRPCHelper.java:69)
 ... 4 more
Caused by: java.io.IOException: Server returned HTTP response code: 503 for
URL: http://someserver.example.com:80/cobbler_api
 at
sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.jav
a:1236)
 at redstone.xmlrpc.XmlRpcClient.handleResponse(XmlRpcClient.java:420)
 ... 7 more
```

Isso indica que houve um problema do Cobbler comunicando com o serviço **taskomatic**. Tente checar o seguinte:

- Verifique se **httpd**está sendo executado: **service httpd status**
- Verifique se o **cobblerd** está sendo executado: **service cobblerd status**
- Verifique que não há regras de firewall que preveniriam conexões **localhost**

5.4. Registro

P: **O comando rhnreg_ks está falhando quando eu o uso, dizendo ERROR: unable to read system id (incapaz de ler o id do sistema). Qual é o problema?**

R: No final do arquivo de kickstart, há uma seção **%post** que registra a máquina ao RHN Satellite:

```
# begin Red Hat management server registration
mkdir -p /usr/share/rhn/
wget http://satellite.example.com/pub/RHN-ORG-TRUSTED-SSL-CERT -O
/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
perl -npe 's/RHNS-CA-CERT/RHN-ORG-TRUSTED-SSL-CERT/g' -i /etc/sysconfig/rhn/*
rhnreg_ks --serverUrl=https://satellite.example.com/XMLRPC --
sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --activationkey=1-
c8d01e2f23c6bbaedd0f6507e9ac079d
# end Red Hat management server registration
```

Interpretar isto na ordem que foi adicionado, ele irá:

- Criar um diretório para acomodar o cert SSL padronizado utilizado pelo RHN Satelite.
- Pegue o certificado SSL para usar durante o registro.
- Busque e substitua as sequências de certificado SSL do arquivo de configuração **rhn-register**, e então registrar ao RHN Satellite, usando o certificado SSL e a chave de ativação. Todo perfil de kickstart inclui uma chave de ativação que certifica que o sistema está atribuído com os canais base e filhos corretos, e receba os direitos de uso do sistema corretos. Se for um reprovisionamento de um sistema existente, a chave de ativação certificará também que está associada com o perfil de sistema anterior.

Se o comando **rhnreg_ks** falhar você poderá ver erros como este no arquivo de log **ks-post.log**:

```
ERROR: unable to read system id.
```

Estes erros também ocorrerão se você tentar realizar um **rhn_check** e o sistema não tiver sido registrado para o RHN Satellite.

A melhor maneira para resolver estes problemas é vizualizar o arquivo de kickstart e copiar e colar os quatro passos diretamente na linha de comando depois que o kickstart estiver completo. Isto produzirá mensagens de erro que são mais detalhadas para lhe ajudar a localizar o problema.

5.5. Kickstarts e Snippets

P: **Qual é a estrutura de diretório para os kickstarts?**

R: O caminho base onde os arquivos de kickstart são armazenados é **/var/lib/rhn/kickstarts/**. Dentro deste diretório, os kickstarts brutos estão no subdiretório **upload**, e os kickstarts gerados pelo assistente estão no subdiretório **wizard**:

```
Raw Kickstarts: /var/lib/rhn/kickstarts/upload/$profile_name--$org_id.cfg
Wizard Kickstarts: /var/lib/rhn/kickstarts/wizard/$profile_name--$org_id.cfg
```

P: **Qual é a estrutura de diretório para snippets do Cobbler?**

R: Snippets do Cobbler estão armazenados no **/var/lib/rhn/kickstarts/snippets**. O Cobbler acessa os snippets usando o link simbólico **/var/lib/cobbler/snippets/spacewalk**.

```
Snippets:  /var/lib/rhn/kickstarts/snippets/$org_id/$snippet_name
```

Importante

Os RPMs do RHN Satelliter supõem que os diretórios kickstart e snippet Cobbler estejam em seus locais padrões, não os mude.

Revision History

Revisão 4-2.4.402	Fri Oct 25 2013	Rüdiger Landmann

Rebuild with Publican 4.0.0

Revisão 4-2.4	Thu Mar 28 2013	Glaucia Cintra

pt-BR translation completed

Revisão 4-2.3	Wed Mar 27 2013	Glaucia Cintra

pt-BR translation completed

Revisão 4-2.2	Mon Mar 11 2013	Glaucia Cintra

pt-BR translation completed

Revisão 4-2.1	Mon Mar 11 2013	Glaucia Cintra_de_Freitas

Arquivos de tradução sincronizados com as fontes XML 4-2

Revisão 4-2	Wed Sept 19 2012	Dan Macpherson

Empacotamento final do 5.5

Revisão 4-1	Thu Aug 9 2012	Athene Chan

Estágio para Lançamento

Revisão 4-0	Mon June 25 2012	Athene Chan

Capítulos atualizados 1 e 2 para o RHN Satellite 5.5
Capítulos 3- 5 editados para o RHN Satellite 5.5
BZ#787348 Linha do iptables do Cobbler iptables incorreta
BZ#702529 Informações adicionais do Kickstart foram adicionadas
BZ#797688 Cobbler Boot ISO não é suportado

Revisão 3-0	Thu May 31 2012	Athene Chan

BZ#826803 - Correção de pontuação na Seção do "Realizando um Kickstart em uma Máquina"
Mudanças gramaticais pequenas na seção do kickstart

Revisão 2-0	Thu May 24 2012	Athene Chan

BZ#783339 - Restruturação de sentença na seção "Provisionando o Troubleshooting Taskomatic"
BZ#783340 - "s390x" to "IBM System z" Atualizado

Revisão 1-3	Mon Aug 15 2011	Lana Brindley

Lançamento do z-stream foi transformado em y-stream

Revisão 1-2	Wed Jun 15 2011	Lana Brindley

Preparado para tradução

Revisão 1-1	Fri May 27 2011	Lana Brindley

Atualizações para tradutores

Revisão 1-0	Fri May 6, 2011	Lana Brindley

Edição final de revisão QE

Preparado para tradução

Revisão 0-8	Thu May 5, 2011	Lana Brindley

BZ#701822 - Revisão QE

Revisão 0-7	Thu April 14, 2011	Lana Brindley

Feedback de revisão ténica

Revisão 0-6	Wed March 23, 2011	Lana Brindley

Preparação para revisão técnica

Revisão 0-5	Tue March 22, 2011	Lana Brindley

BZ#666435
BZ#666846
BZ#678080
BZ#682364

Revisão 0-4	Tue March 22, 2011	Lana Brindley

Solução de problemas

Revisão 0-3	Mon March 21, 2011	Lana Brindley

Satellites múltiplos

Revisão 0-2	Thu March 17, 2011	Lana Brindley

Introdução
Kickstart
Comando Avançados
Reestruturação de alguns capítulos

Revisão 0-1	Wed Jan 5, 2011	Lana Brindley

Completas novas estruturações de capítulos

Revisão 0-0	Tue Dec 21, 2010	Lana Brindley

Nova criação de documento do Guia de Implantação do RHN Satellite original